WHAT IS CLAIMED IS:

1. A serverless name resolution protocol through which unique numbers are resolved to addresses, comprising the steps of:

receiving at a first node a request message from a requester node seeking address resolution of a second node having a unique number identifier, the request message including address information of the requester node;

populating a routing table of the first node with the address information of the requester node;

analyzing the request message;

generating a response message to the requester node identifying address information of the first node as best matching for the request message when one of three conditions is met; otherwise

determining a suitable next hop for the request; and

forwarding the request message to the suitable next hop.

2. The protocol of claim 1, wherein the step of analyzing the request message comprises the step of comparing the unique number identifier to the address information of the first node, and wherein the step of generating a response message to the requester node identifying address information of the first node as best matching for the request message when one of three conditions is met comprises the step of generating a response message to the requester node identifying address information of the first node as best matching for the request message when the unique number identifier is identical to the address information of the first node.

3. The protocol of claim 1, wherein the request message contains a maximum hop count value and a list of node that have processed the request message, and wherein the step of analyzing the request message comprises the step of determining if a number of nodes which have previously processed the request message exceeds the maximum hop count, and wherein the step of generating a response message to the requester node identifying address information of the first node as best matching for the request message when one of three conditions is met comprises the step of generating a response

message to the requester node identifying address information of the first node as best matching for the request message when the number of nodes which have previously processed the request message exceeds the maximum hop count.

4. The protocol of claim 1, wherein the request message contains a list of nodes that have processed the request message, and wherein the step of analyzing the request message comprises the step of determining if the address information of the first node is in the list of nodes that have processed the request message, and wherein the step of generating a response message to the requester node identifying address information of the first node as best matching for the request message when one of three conditions is met comprises the step of generating a response message to the requester node identifying address information of the first node as best matching for the request message when the address information of the first node is in the list of nodes that have processed the request message.

5. The protocol of claim 1, wherein the request message includes a certificate of origin, further comprising the steps of checking the certificate of origin to determine its validity, and refusing the request message when the certificate of origin is invalid.

6. The protocol of claim 1, wherein the step of populating the routing table comprises the steps of:

determining if the address information of the requester node is already in the routing table;

refreshing the address information of the requester node if more recent than the address information of the requester node already stored in the routing table; else

computing the distance between the address information of the first node and the requester node;

determining from the distance a selected level into which to store the address information of the requester node; and

storing the address information in the selected level.

7. The protocol of claim 6, wherein the selected level is a last level having K entries stored therein, and wherein the step of determining the selected level comprises the steps of determining that an entry should be replaced, and replacing the entry with the address information of the requester node.

8. The protocol of claim 6, wherein the selected level is a last level, further comprising the steps of preparing a flooding message containing the address information of the first node with an empty list of already flooded nodes, and sending the flooding message to the requester node.

9. The protocol of claim 8, further comprising the steps of preparing a list of nodes in the routing table whose distance to the requester node is smaller than $DMAX/(P^{(L-1)})$, remove from the list nodes that are marked as already flooded when the addition of the new entry is a result of a flooding message, preparing a flooding message containing the address information of the requester node, and send the flooding message to all nodes in the list.

10. The protocol of claim 6, wherein the selected level is a last level having K or more entries stored therein, and wherein the step of determining the selected level comprises the step of adding a new level and splitting the entries in the last level between the last level and the new level according to a distance from the address information of the first node.

11. The protocol of claim 1, further comprising the steps of checking a date of validity for address information in the routing table, and removing address information for which the date of validity has passed.

12. The protocol of claim 1, wherein the step of determining a suitable next hop for the request comprises the steps of finding a subset of routing table entries whose address is not already listed in the request message, returning an indication of failure when the subset is empty, returning a particular entry when the particular entry is the only entry in the subset.

13. The protocol of claim 12, further comprising the steps of finding two entries whose identifiers are closest to the second node, randomly pick one of the two entries, and return the randomly picked entry.

14. The protocol of claim 1, further comprising the steps of:

receiving a response message including address information of the second node and address information of a best match node;

comparing the address information of the second node and the address information of the best match node;

replacing the address information of the best match node with the address information of the first node when the address information of the best match node is not equal to the address information of the second node and the address information of the first node is closer to the address information of the second node than the address information of the best match node; and

relaying the response message to the requester node when the requester node is not the first node.

15. The protocol of claim 1, further comprising the step of forming the unique number identifier of the second node by computing a hash of a name of the second node.

16. The protocol of claim 15, wherein the step of forming the unique number identifier further comprises the step of associating a unique number with the hash of the name to form the unique number identifier in the form <hash>.<unique number>.

17. The protocol of claim 1, further comprising the step of extracting the unique number identifier of the second node from a unique name processed through a DNS query to a peer to peer server, the unique name taking the form <peer to peer identifier>.<DNS server address>.

18. The protocol of claim 17, wherein the <peer to peer identifier> is a unique name, further comprising the step of forming the unique number identifier of the second node by computing a hash of a name of the second node.

19. The protocol of claim 18, wherein the step of forming the unique number identifier further comprises the step of associating a unique number with the hash of the name to form the unique number identifier in the form <hash>.<unique number>.

20. A multilevel cache for use in a peer to peer name resolution protocol, comprising a set of L levels, each level sized to hold K entries, and wherein the number of levels L is dynamic, increasing by 1 when a K+1 entry is to be added to a L^{th} level when the L^{th} level is occupied by K entries.

21. A method of dynamically maintaining a multilevel cache for use in a peer to peer name resolution protocol, comprising the steps of:

receiving a new entry to be entered into the multilevel cache;

determining in which level the new entry is to be stored;

determining if the level into which the new entry is to be stored is full;

storing the new entry in the level when the level is not full;

randomly replacing another entry with the new entry when the level is full and when the level is not a last level of the multilevel cache; and

adding a new level when the level is full and when the level is the last level of the multilevel cache, sorting entries from the last level between the last level and the new level, and storing the new entry.

22. The method of claim 21, wherein the step of determining in which level the entry is to be stored comprises the step of determining if the entry already exists in the multilevel cache, and replacing an address certificate for the entry when the address certificate is more recent than the address certificate of the entry stored in the multilevel cache.

23. The method of claim 21, wherein the step of determining in which level the new entry is to be stored comprises the step of computing a distance between a local identifier and an entry identifier.

24. A method of resolving a peer name to a peer address certificate, comprising the steps of computing a hash value of the peer name, associating therewith a unique number in the form <hash>.<unique number>, formatting a request message, and transmitting the request message to a peer node.

25. The method of claim 24, further comprising the steps of:

receiving a response message including the peer address certificate and address information of a best match node;

comparing the peer address certificate and address information of a best match node;

replacing the address information of the best match node with the address information of a local node when the address information of the best match node is not equal to peer address certificate and the address information of the local node is closer to peer address certificate than the address information of the best match node; and

relaying the response message to the requester node when the requester node is not the local node.